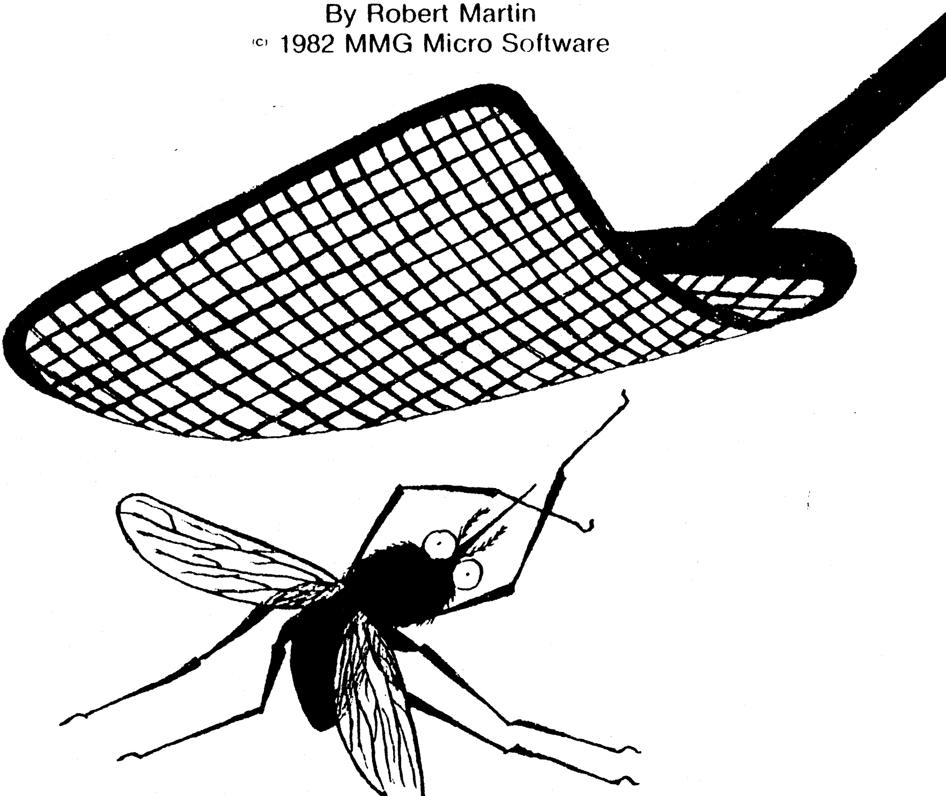
MMG BASIC DEBUGGER

By Robert Martin



REQUIRES 16K RAM • ATARI 400/800
810 DISK DRIVE • BASIC CARTRIDGE

MMG BASIC DEBUGGER
by Robert Martin

Congratulations! You have just purchased THE program which helps you solve the hardest part of BASIC programming for your ATARI home computer, debugging your program after writing it. MMG BASIC DEBUGGER is a set of tools that will let you look inside a BASIC program while it is being executed, follow the path of the program, examine variables and modify the program during a run. Some features of MMG BASIC DEBUGGER are TRACE (many options), SEARCH, SPLIT SCREEN, SCROLLING two parts of your BASIC program simultaneously, CROSS REFERENCING of variables, and two display screens. Each feature has several options accessed by one or two letter commands. To fully master this collection of programs, a thorough reading of this documentation is necessary. The easiest way to familiarize yourself with the operation of this debugger is to boot-up your computer with the MMG BASIC DEBUGGER disk, and then LOAD "D:DEMO", a BASIC program included on your disk. The instructions below can then be tested as they are described.

****

WARNING

DO NOT ATTEMPT TO WRITE TO YOUR MMG BASIC DEBUGGER DISK!

DO NOT REMOVE THE WRITE-PROTECT TAB FROM THE DISK

EITHER OF THESE ACTIONS WILL VOID YOUR WARRANTY

ALWAYS HAVE A BACKUP COPY OF THE PROGRAM YOU ARE DEBUGGING ON ANOTHER DISK!!

****

GETTING STARTED

Turn on your disk drive, and wait for the top red light to go out. Insert the MMG BASIC DEBUGGER disk, and close the door of the disk drive firmly. Check to be sure that your ATARI BASIC cartridge is correctly inserted in your computer, and close the cover securely. Turn on your monitor or TV, and then turn on your ATARI computer. MMG BASIC DEBUGGER will automatically load into your computer's memory. When the drive has stopped, REMOVE the MMG BASIC DEBUGGER disk, and put it in a safe place. Insert a disk containing the BASIC program you wish to debug (BE SURE YOU HAVE A BACKUP OF THE PROGRAM ON ANOTHER DISK), and type LOAD "D:filename.ext", where, in place of filename.ext, you type the name of your BASIC program. To follow our examples, the file name to use is "D:DEMO". Then hit the RETURN key.

Throughout this documentation, reference to CONTROL-Z, for example, means that you hold down the CTRL key while hitting the letter Z. CONTROL-T then refers to holding down the CTRL key while hitting the letter T. Reference to SHIFT-CONTROL-D means that you hold down both the SHIFT and CTRL keys together, while hitting the letter D.

AUXILIARY SCREEN

The first feature you should become aquainted with is the auxiliary screen. This screen is accessed by pressing CONTROL-Z (or D during trace). The primary use of the auxiliary screen is to follow the progress of your program during trace, but other uses are described on the next page.

Use it to serve as a work space. Examples of such uses include:

- calculations
- writing a subroutine while retaining a listing on the regular screen, toggling back and forth for reference
- modifying your program, by adding or deleting lines, while retaining the main listing on the regular screen
- storing your data for quick reference, such as variables, listed lines, etc.
- executing commands, such as ENTER "D:filename.ext" to merge a program with your current one
- a general scratch pad, but, in contrast to the ATARI memo pad mode, all normal BASIC functions are still in effect

The auxiliary screen is totally independent of the regular screen, and will not accept graphic commands, or attempts to change graphic modes. However, POKEs to locations that affect the display will be reflected in this screen, if it is being viewed when those POKEs take effect. For this reason, TRACE will be halted when you toggle to the auxiliary screen, to prevent this from accidentally happening during the execution of a line.

Now let's access the auxiliary screen. Type CONTROL-Z (remember, hold down CTRL while hitting Z). The color of the screen border changed in the auxiliary screen, so that you can tell which screen you are viewing. The top of your screen should now read Auxiliary Screen as well, to identify the mode.

CAUTION: DO NOT change screens while BASIC is writing to the screen; your computer may crash!!

Type CONTROL-Z again, to return to the regular screen. Let's put something on the regular screen; type LIST 1,100, and RETURN. The screen will fill with the first few lines of DEMO. Now type CONTROL-Z, and notice that the auxiliary screen is still clear. Let's do a quick calculation here. Type PRINT 256*PEEK(561) + PEEK(560) and RETURN. This should print out the location of the display list, which will depend on the amount of memory in your ATARI. Hit CONTROL-Z again, just to see that your listing is still on the regular screen. Hit CONTOL-Z again, and your calculation is still there, waiting for you. We can now clear this screen, by hitting ESC, followed by SHIFT-CLEAR. Your regular screen is still there (check if you don't believe it), but your auxiliary screen is now cleared for further action. Type LIST 101,200 and you will list the second part of DEMO to your auxiliary screen. You can now toggle back and forth between the two screens with CONTROL-Z. However, there is a much better way to do this, and we'll try that next.

SPLIT SCREEN

Let's first clear both screens. Hit ESC and then SHIFT-CLEAR to clear the auxiliary screen, and then CONTROL-Z, followed by ESC and SHIFT-CLEAR to clear the regular screen. Now, to view the bottom of the regular screen and the top of the auxiliary screen at the same time, press SHIFT-CONTROL-D (all 3 keys simultaneously). The screen will be split by a horizontal white bar, and the lower half of the screen will have a light border, signifying the auxiliary screen. Now type LIST 1,100 and watch

what happens. The listing scrolls by on the top window, and what remains visible is the lower part of the listing. Now type CONTROL-Z, and then LIST 101,200 and you should see the same thing happen in the bottom window. NOTE: since you are viewing only half of each screen, it is possible that you will not see the cursor in either window, since it may very well be in a portion of the screen which you cannot see. If this happens, use the cursor control keys to move it to a visible area again.

You can still edit each screen in this mode. CONTROL-Z toggles between the two screens, and any changes you make in a program line in either window will be incorporated into your program. To clear either screen, put the cursor in that screen, and type ESC followed by SHIFT-CLEAR.

To exit from the split-screen mode, type SHIFT-CONTROL-D a second time. Then let's clear the screen with ESC followed by SHIFT-CLEAR, and we'll move on to the next function.

SCROLL

The scrolling feature is accessed by typing either SHIFT-CONTROL-↑ or SHIFT-CONTROL-↓. When these commands are given, the prompt # will appear on the first free line above or below the lines currently listed on your screen. In this example, your screen was clear, so you will now see the # symbol. Type 100 and RETURN, and line 100 will appear on the screen. The default (a RETURN without a number) will start with the next line number after the one already on the screen.

To continue scrolling up or down simply press the up arrow or down arrow (without pressing the

CONTROL key). Press the up arrow, and you will see additional lines of DEMO appear on the screen. Press the down arrow, and you will see the scroll go the other way. You can scroll through your whole program, line by line, very rapidly this way. If you scroll off the end of your program, the prompt NO MORE LINES appears on the screen in place of the expected line. No harm done-just start scrolling the other way. When you find a part of your program you want to change, just hit any key other than the up or down arrows to exit the scroll mode, and make whatever changes you'd like to make.

This scrolling mode also works in the split screen mode. If you'd like to try it, enter the split screen mode with SHIFT-CONTROL-D, and then enter the scroll mode with SHIFT-CONTROL-↑ or SHIFT-CONTROL-↓ . Enter the line number and RETURN, and scroll with the up or down arrow keys. Exit the scroll mode by hitting any other key, shift screens with CONTROL-Z, re-enter the scroll mode in this window with the SHIFT-CONTOL-arrows, type a line number and RETURN, and scroll this window independently of the other window. This gives you the ability to simultaneously edit two parts of your program, such as a calling routine and the subroutine it calls. This is a very powerful feature, and should be explored thoroughly on your own, to master it.

TRACE

The fundamental tool of MMG BASIC DEBUGGER is the TRACE program, which allows you to trace the execution of a program line-by-line. This feature

is fundamental to any BASIC debugger, and for this reason has the most number of options associated with it. To initiate the TRACE option, press CONTROL-T. The prompt ENTER TRACE OPTION will appear, allowing you to select the options you would like to use. First, let's do a standard TRACE. At the prompt, type RETURN, to get the default options. On your regular screen, the one you are viewing now, you will see the program DEMO being executed. Let it proceed for a bit, and then hit S to stop the TRACE. Now type CONTROL-Z to look at the auxiliary screen. AHAH! What you should see is a series of numbers, arranged neatly in columns. These are the line numbers of DEMO, arranged in the order that they were executed. NOTE: not the order that they appear in the program, but the order in which the program flow executed them. By reviewing this list of line numbers, it is easy to TRACE the execution of your program, to determine whether or not it is performing the way you expected. Note that you may execute TRACE while viewing the auxiliary screen rather than the regular screen, if you simply want to see the line numbers. Try it! Type CONTROL-T, and at the prompt, type RETURN. You will see the line numbers appear as BASIC executes your program. Stop the listing with S, and toggle back to the regular screen with CONTROL-Z, to see the output of DEMO up until the point at which you stopped it.

Many different options are available for this TRACE function, and they are briefly described on the following two pages. After the list of the available options, we will be discussing and using them in examples, but for now, just review them.

OPTION	DESCRIPTION
RETURN	The default option is obtained by typing a RETURN without entering any options. This will TRACE starting with the first line and list the line numbers executed to the auxiliary screen.
nnn	TRACE starting at line number nnn. The number must be at the beginning of the option line. Line numbers executed are listed to the auxiliary screen during TRACE if no other options are selected.
Rnnn	Run to line number nnn, and then begin TRACE. This command allows you to get past parts of the program that are already debugged, or to get by initialization routines. This option must be at the beginning of the option line, and it may not be used with the nnn option above.
S	In STEP TRACE mode, one line is executed each time the space bar is pressed. You can toggle between STEP TRACE and TRACE by pressing T at any time during a TRACE.
L	LISTs executed lines to the auxiliary screen. This option is slower than just listing the line numbers, but it does allow you to examine each line as it is executed.
V	This option prints the VALUEs of up to five variables (scaler, array or string variables are allowed) to the auxiliary screen as you

TRACE. After the command line has been chosen, MMG BASIC DEBUGGER will request each variable, one at a time. After typing in a variable name, hit RETURN and you will be asked for the next variable name. If you don't wish to enter any more variables, just hit RETURN.

W TRACE WHILE will cause the TRACE to change to STEP TRACE when a condition is no longer true. This option allows you to test the value of a scaler variable (array or string variables are not allowed) against a value and conditions of your choice. Test conditions allowed are greater than, less than, equal to, or not equal to. After the TRACE options have been selected, you will be asked for the variable, the condition, and the test value before beginning the TRACE. For example, you could decide to TRACE WHILE XSET was less than 3. When XSET was greater than or equal to 3, TRACE would change to STEP TRACE, allowing you to single step through your program from that point on.

C CONTINUE where you left off. If you temporarily stopped the TRACE to change a variable or enter a new line to your program, this command allows you to restart the program where you left off.

P PRINT all output (GRAPHICS 0) to the printer.

PA PRINT only the AUXILIARY screen output to the printer.

IMPORTANT NOTE: the TRACE function will NOT trace through the following BASIC commands: RUN, LOAD, STOP and LIST. It also requires line 0 for its own use, so if you have a line 0, renumber it before TRACEing your program.

Let's try some examples of the different TRACE modes. We have already seen the default option, which prints the line numbers of the executed lines to the auxiliary screen. This is accessed from the TRACE program by just hitting RETURN in response to the prompt. Sometimes, however, you would rather see the whole BASIC line, rather than just the line number, as each line is executed. To accomplish this, type L at the prompt, and RETURN. The TRACE will start. Let it go for a few seconds, and stop it with the S key. Then toggle to the auxiliary screen with CONTROL-Z, and you will see each line of the DEMO program listed there, in the order that they were executed!

Since the TRACE option just proceeds through your program line-by-line, wouldn't it be nice to be able to single-step through, rather than all at once? You can! Re-enter the regular screen with CONTROL-Z, and the TRACE option with CONTROL-T. Respond to the TRACE prompt with S, and then hit RETURN. The first line of DEMO will be executed, and the computer will stop, waiting for you to hit the space bar. When you do, the next line in the program flow will be executed. Note that this is not necessarily the second line of your program, but will be the second line executed. Type S now, to stop the TRACE, and toggle to the auxiliary screen to see the line numbers of the lines executed. We will now introduce a new option, C (for Continue). Let's type CONTROL-T to enter the

TRACE mode while we watch the auxiliary screen. In response to the prompt, type CS and hit RETURN. To execute more than one option of the TRACE program at a time, just type in the option codes, in any order, all on a line, with no spaces, and follow them with a RETURN. The auxiliary screen will now list a new line number each time you hit the space bar, allowing you to single-step through your program, while watching either the program's output on the regular screen, or the program flow, on the auxiliary screen. We could have typed CSL if we had wanted to see the whole BASIC line, rather than just the line numbers, on the auxiliary screen.

Any of the options can be made to begin at some place in your program other than at the beginning, simply by starting your response to the TRACE prompt with a line number. For instance, 100S will begin STEP TRACE at line 100, or 200 just followed by RETURN will start a regular TRACE at line 200. Often, however, the beginning of your BASIC program has certain start-up or set-up routines which are required for some later part to function correctly. Beginning the TRACE in the middle of such a program will not work, so MMG BASIC DEBUGGER has the R (RUN) option provided for such cases. If you type R100SL, for example, DEMO will run to line 100, and then enter the STEP TRACE mode, printing entire BASIC lines to the auxiliary screen. Other combinations of commands using the RUN option should be explored for full understanding of this mode.

The V option of TRACE is particularly useful when you think that your program is malfuctioning because some variables are being incorrectly assigned values. To demonstrate this option, enter

TRACE and type VS in response to the prompt, for the Variable option STEP TRACE. The screen will then ask you for a variable name. Type LINE, and hit RETURN. You will then be asked for a second variable name, and so on, up to five, if you so desire. Type I for the second variable, and X for the third. At the fourth request, just type RETURN, which will begin the TRACE. Hit the space bar 10 times, and stop the TRACE by hitting the S key. Toggle to the auxiliary screen. You should see the names of the variables you selected, LINE, I, and X, printed on the top of the screen, and the values of each of the variables printed in columns, with each row representing the values of the variables following the execution of one line of DEMO. This option eliminates the need for PRINT statements all through your program to tell you what values the variables have at any point!

If you have a printer, you'll appreciate the P and PA options of TRACE. Include P on the command line following the TRACE prompt, and all GRAPHICS 0 output will be printed on your printer, for a permanent record. PA will print only the output normally directed to the auxiliary screen to your printer. Neither of these options will have any effect on the output to either screen. Note that if your printer is not on, or you don't have a printer connected, the TRACE proceeds anyway. The lack of a printout should be a strong clue to the problem!

Finally, the last option is W, for TRACE WHILE. This mode is a conditional TRACE. In response to the TRACE prompt, type WL, which will enter the TRACE WHILE mode, listing whole BASIC lines to the auxiliary screen. The screen will

prompt you for the conditional variable; that is, the variable upon which you would like to make the TRACE dependent. Type LINE, and follow it by a RETURN. You will then be asked for the condition; that is, greater than, less than, equal to, or not equal to. Type the greater than (>) key, and RETURN. Finally, you will be asked for the value you wish to test against. Type 200 and RETURN. What you have done is to tell the computer to TRACE until the variable LINE is greater than 200, and then to change to STEP TRACE, all the time listing entire BASIC lines to the auxiliary screen. Experiment with this option, and you will find it to be extremely valuable in debugging your programs.

KEYBOARD INPUT

During TRACE, the keyboard is disabled except for the S, D and T keys. The S key halts TRACE, the D key allows you to toggle between the regular screen and the auxiliary screen while in TRACE mode, and the T key toggles between TRACE and STEP TRACE. The normal function of the keyboard can be enabled by pressing CONTROL-ESC. Another CONTROL-ESC toggles back to the disabled keyboard. This toggling is automatic if your BASIC program needs keyboard input. Should you ever find your keyboard unresponsive to input, try hitting CONTROL-ESC to re-enable it.

SEARCH

You can SEARCH a program for any sequence of characters, strings, BASIC statements or commands, by pressing SHIFT-CONTROL-S (you must hit all 3

keys simultaneously). The prompt ENTER STRING FOR SEARCH will appear. Enter the string you wish to search for, and press RETURN. The prompt LINE NUMBERS FOR SEARCH will next appear, allowing you to choose the block of lines to be searched. Just pressing RETURN at this point will cause the search to be conducted for all lines in your BASIC program. After typing RETURN, the search will begin. To search a 200-line BASIC program will require about 5 seconds.

CROSS REFERENCE

You can also produce a listing of a cross-reference table for the variables in your program, by typing SHIFT-CONTROL-R (all 3 keys simultaneously). The program will ask you if you want the output directed to a printer (in addition to the screen); if you type Y the output will also appear on your printer. Any other key causes output to be directed only to the screen.

The CROSS REFERENCE lists all variables in your program in alphabetical order with the line numbers on which they occur listed with them. CONTROL-1 will temporarily halt the listing for you to inspect it, and a second CONTROL-1 will restart it again. Exit this mode by hitting the BREAK key.

BREAK KEY AND SYSTEM RESET

The BREAK key and SYSTEM RESET key are enabled during a TRACE but use only the S key, if at all possible. The S key will stop the program under the most controlled conditions and the program can be restarted (using the C option) from the same

place. However, if the BREAK key is used, the C option in TRACE will not start the program again.

SYSTEM RESET does what it wants to do at all times, and if you press SYSTEM RESET during a TRACE, your program may be altered, giving you an opportunity to load from your back-up copy. (You did have a back-up copy, didn't you? We warned you!) Even hitting SYSTEM RESET will not affect MMG BASIC DEBUGGER, however. It will remain there for your use, whenever you need it.

We very much hope you enjoy MMG BASIC DEBUGGER, and that it makes your life easier, as it has ours. If you have comments or suggestions, we'd love to hear them. Please write and let us know.

LIMITED WARRANTY

This software product is sold "AS IS", without warranty as to its performance. The entire risk as to the quality and performance of the computer software program is assumed by the user. The user, and not the manufacturer, distributor or retailer assumes the entire cost of all necessary service or repair to the computer software program.

The above warranty is in lieu of all other express warranties and of implied warranties or merchantability and fitness for a particular purpose or any other warranty obligation on the part of MMG Micro Software shall be limited to replacement of the original software program, should it become defective within the first ninety (90) days of use by the original purchaser. Some states do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you. In no event shall MMG Micro Software or anyone else who has been involved in the creation and production of this computer software program be liable for indirect, special, or consequential damages, such as, but not limited to , loss of anticipated profits or benefits resulting from the use of this program, or arising out of any breach of this warranty. Some states do not allow the exclusion or limitation of incidental or consequential damages so the above limitation may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

The user of this product shall be entitled to use the product for his/her own use, but shall not be entitled to sell or transfer reproductions of the product or instructional materials to other parties in any way.